

# Physical computing mit Scratch und scratchClient **Anfänger**

*Raspberry Pi, Arduino, Scratch und  
scratchClient steuern Servos, LEDs und mehr*

Hans de Jong & Gerhard Hepp

Workshop für die Pi And More und Maker Faire  
Konferenzen

# Teil 1: Einleitung

# On-line Videos

- Für on-line Videos, gehe zu Youtube und suche *scratchClient Tutorials*.
  - Videos sind auf Englisch.
  - Diese Präsentation ist im Details beigearbeitet, aber die Video kann noch immer benutzt werden.
- ScratchClient hat zwei Weisen um Blocken zu representieren:
  - Workshop Mode: benutze Nahm-Werte Paaren
  - Verbose Mode: jede Input und Output hat sein eigenes Block.
- In diesen Workshop benutzen wir Workshop Mode
  - Sehe Appendix A wie man eine andere Representation bekommen kann, die oft einfacher ist.

# Workshop-Organisation

- Begrüßung und Einführung Präsentation (5 min).
  - Danach wird jeder in seinem eigenen Tempo arbeiten.
  - Wählen Sie mit Ihrem "Arbeitsplatzpartner", welche Themen Sie bearbeiten möchten.
    - Es gibt mehr Material, als in 110 Minuten abgedeckt werden kann.
  - Sprache: Deutsch, Englisch und Holländisch
  - Am Ende kopieren Sie das von Ihnen erstellte Material auf Ihren USB-Stick (wenn Sie möchten)
  - Aufbrechen und aufräumen (5 min)
- Die wichtigsten Schritte:
    1. Holen Sie sich eine funktionierende Hardware- und scratchClient-Konfiguration mit einem scratchClient-Konfigurationstool.
    2. Stellen Sie Komponenten auf die Platine und testen Sie Ihr Setup.
    3. Schreibe etwas Code in Scratch
    4. Fügen Sie weitere Hardware hinzu und aktualisieren Sie die Konfigurationsdatei.
    5. Und wiederhole das.

# Ziele

- Am Ende des heutigen Tages sollten Sie in der Lage sein, einige dieser Dinge zu tun:
  - Reproduzieren Sie das Setup zu Hause (vorausgesetzt, Sie haben die Hardware)
    - Sie können alle Dateien über <https://github.com/hansdejongehv/scratchClient-Tutorials> abrufen
    - Oder gehen Sie zu [www.github.com](http://www.github.com) und suchen Sie nach *scratchClient*
  - Verstehen (abhängig davon, wie weit Sie kommen und wie tief Sie tauchen)
    - Digitalausgang (z. B. Beleuchtung einer LED)
    - Digitaler Eingang (z. B. Tastenerkennung)
    - Analogeingang (z. B. von einem Potentiometer)
    - Pulsbreitenmodulation (PWM)
      - Zum Dimmen von LEDs
      - Zur Steuerung von Servos
      - Um einen Summer zu hören
  - Verstehen Sie, wofür alle Widerstände sind
  - In der Lage sein, scratchClient zu konfigurieren und auszuführen
  - Ein Scratch Programm zu schreiben zur Steuerung der physikalischen Ein- und Ausgabe
  - Überwachung der Ein- und Ausgänge
- **Habe Spaß!**

# Keine Ziele

- Es ist kein Ziel, ein komplettes nützliches Spiel oder ein anderes Programm zu erstellen.
  - Sie können dies mit Ihrer eigenen Kreativität zu Hause tun, da Sie jetzt wissen, wie Sie mithilfe von scratchClient mehrere Hardwareteile von Scratch aus steuern können.

# Beispiel für was mit Scratch und scratchClient erstellt werden kann

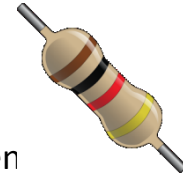
- <https://www.youtube.com/watch?v=Qo1gnXNzhqE>

# Versionen von Scratch

- scratchClient kann damit arbeiten
  - Scratch 1.4
  - Der (für Raspberry Pi) neue Scratch 2
- Dieser Workshop ist für Scratch 2 geschrieben
  - Scratch 2 auf Raspberry Pi hat einige Bugs, aber wir werden es umgehen.
  - Wenn Sie sehen möchten, wie man es für Scratch 1.4 macht, siehe das Ende der Präsentation in Anhang B.



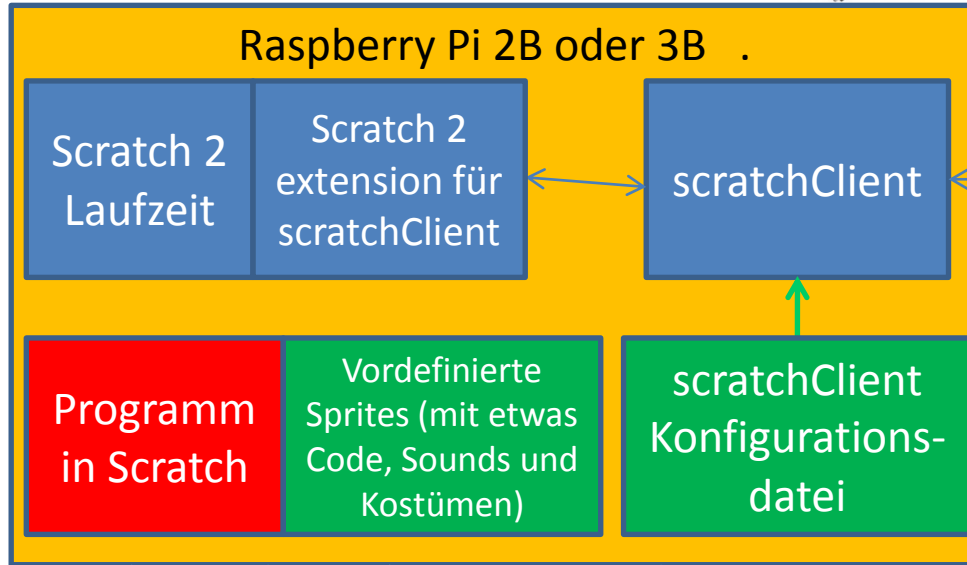
# Nur ein paar Regeln heute



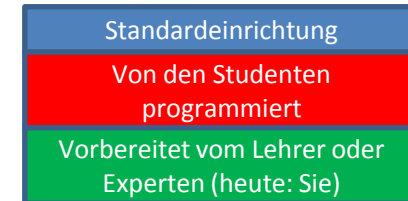
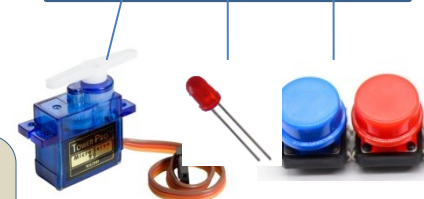
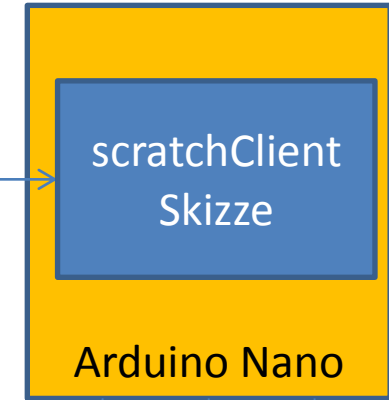
- Setzen Sie immer einen Widerstand in Reihe mit den Komponenten, wenn angegeben
  - Wenn Sie denken, dass es keine Notwendigkeit gibt, dann sagen Sie es uns bitte und wir werden es erklären was der Grund ist.
- Beim Ändern der Verdrahtung
  1. Lösen Sie das USB-Kabel vom Arduino Nano
  2. Lösen Sie die 9V-Stromversorgung aus
  3. Überprüfen Sie, überprüfen Sie nochmals und überprüfen Sie erneut, ob die Verkabelung korrekt ist. Sie können Bauteile bei falscher Verdrahtung sprengen!
  4. Stellen Sie sicher, dass Sie **beide** (4-Augen-Prinzip) davon überzeugt sind, dass die Verkabelung in Ordnung ist bevor Sie das Gerät wieder anschließen und wieder einschalten.
  5. Nach dem Ändern einer Konfigurationsdatei: starte scratchClient neu
- Wenn etwas kaputt geht oder beschädigt wird: wir haben etwas Ersatzmaterial
  - Bitte legen Sie nichts was kaputt ist zurück in die Schachtel.



# Die Einrichtung



Definiert die Funktion der GPIO-Pins auf dem Arduino (und RPi) und gibt ihnen logische Namen.



# Oft gefragt: warum einen Arduino verwenden?

- Frage: warum nicht normal die GPIO Pennen der Raspberry Pi verwenden?
- Antwort:
  - Arduino verfügt über Analog- und Hardware-Pulsweitenmodulation (PWM) für eine stabilere Servosteuerung.
  - Arduino verwendet 5 Volt. Raspberry Pi GPIO Stifte sind nicht 5 Volt tolerant.
  - Wenn Kinder Fehler mit der Elektronik machen, werden sie möglicherweise einen Arduino Nano-Klon von 2 Euro anstelle eines Raspberry Pi von 35 Euro sprengen.
  - Es macht das Setup transportabel, jetzt muss nur ein einziger USB-Stecker angeschlossen werden und kein großes Kabel an den GPIO-Pins.

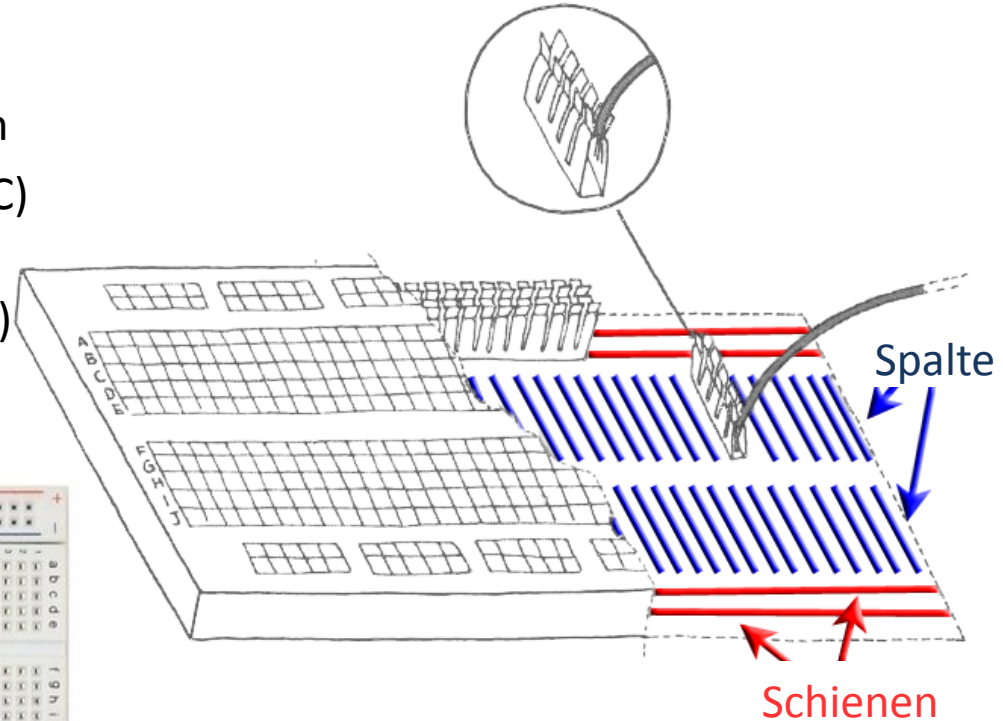
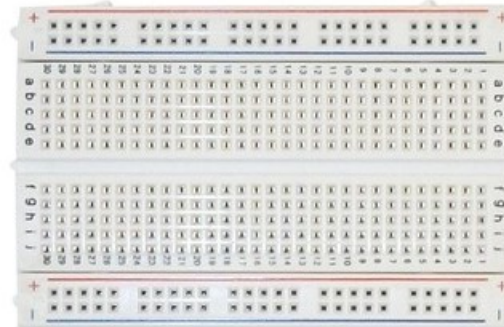
# Wählen Sie, was Sie tun möchten

- Der Workshop ist für alle da, vom Anfänger bis zum Experten.
- Es gibt nicht genug Zeit, um alles zu tun, also wählen Sie, was Sie tun möchten.
- **Gelbe Folien** haben Hintergrundinformationen und Sie können sie überspringen, wenn Sie möchten oder später schauen
- Empfohlen
  - Alle: Lernen Sie, scratchClient zu konfigurieren mit LED (digital aus) und Knopf (digital ein)
  - Alles: Probieren Sie es in Scratch aus
  - Danach: Wählen Sie weitere Themen aus separaten Dateien
    - Zwischenstufe (Material nur auf Englisch)
    - Fortgeschrittenes Niveau (Material nur auf Englisch)
    - Expertenlevel (Material nur auf Englisch)

# Teil 2: Kennenlernen der Komponenten

# Steckbrett

- Wird verwendet, um schnell elektronische Schaltkreise aufzubauen
- Beachten Sie die 2 Schienen für + (VCC) und - (GND)
- Beachten Sie bei jeder Spalte (vertikal) die 2 Balken mit je 5 miteinander verbundenen Löchern.



# Mit Blick auf die Arduino Nano Adapterplatine

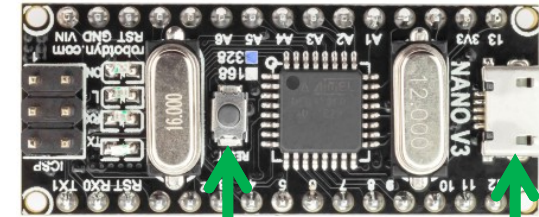
Pro GPIO Signal 3  
Steckerleisten:  
S (blau = Signal)  
V (rot = VCC = +)  
G (schwarz = GND = -)  
(Sehr praktisch für z.B.  
Servos anschließen)

Analoge Ports (die meisten können  
auch als digitale Ports verwendet werden)

Reset Taster

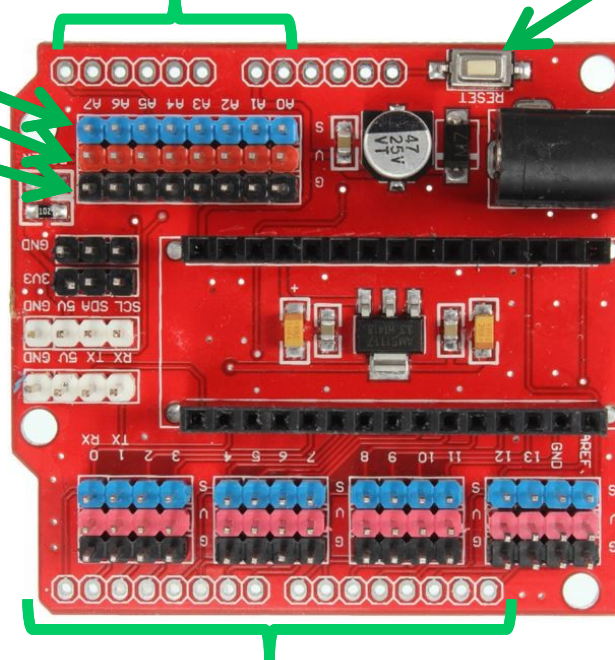
9V Stromansluß

Arduino Nano mit  
328P Prozessor



Micro USB Port

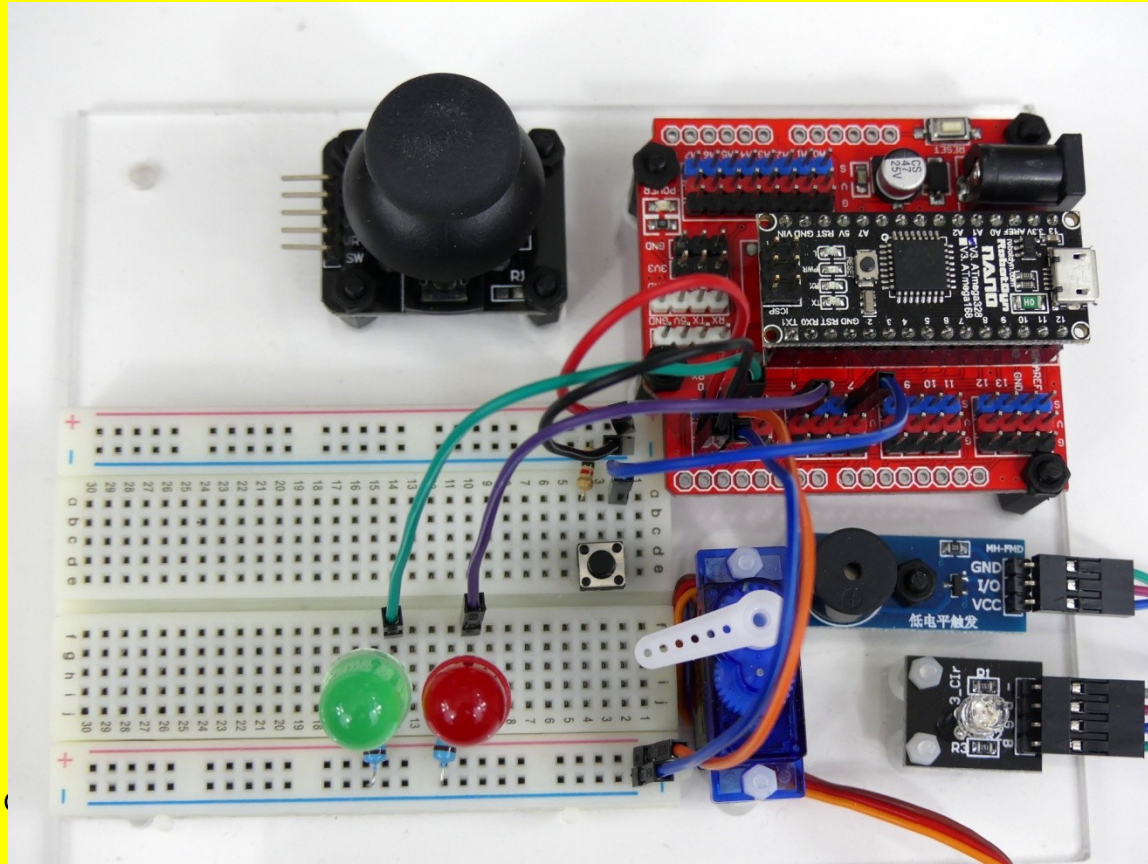
Reset Taster



Digital Ports



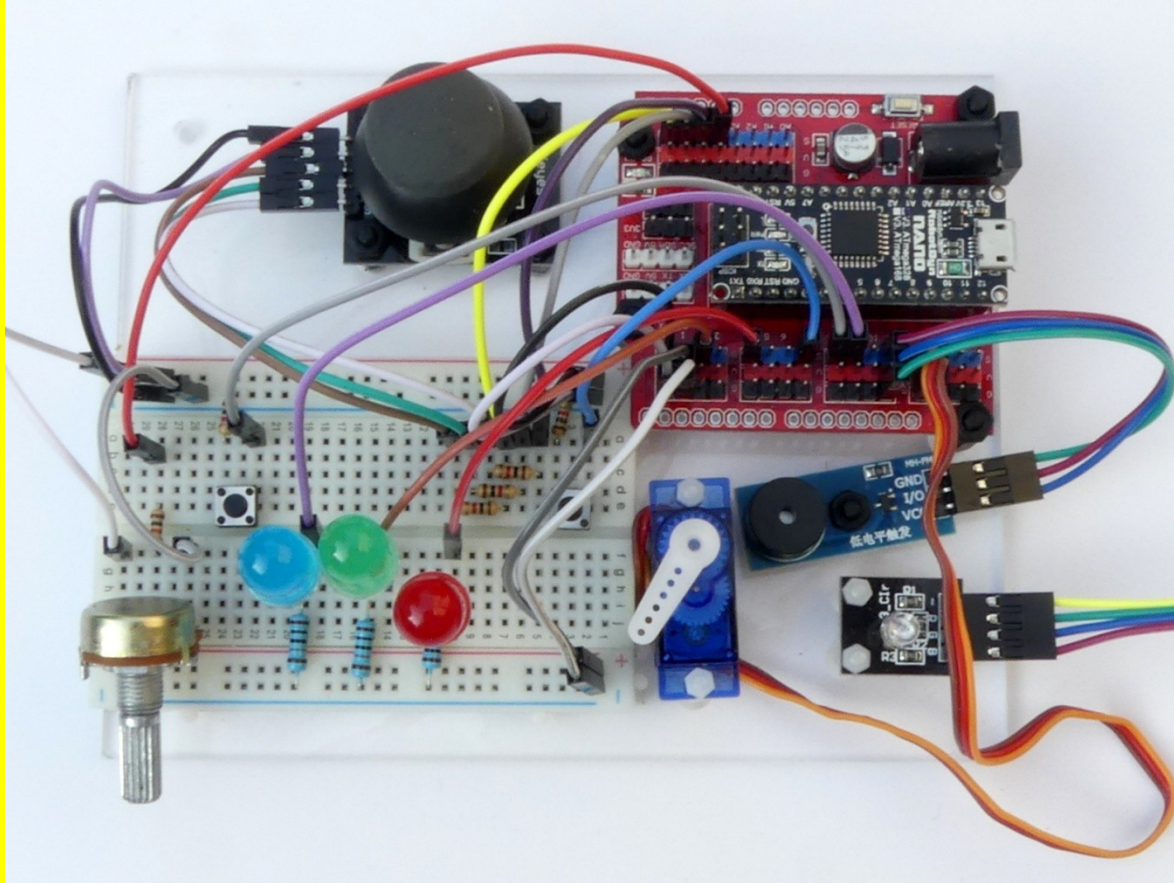
# Wie das letzte Brett aussieht ...



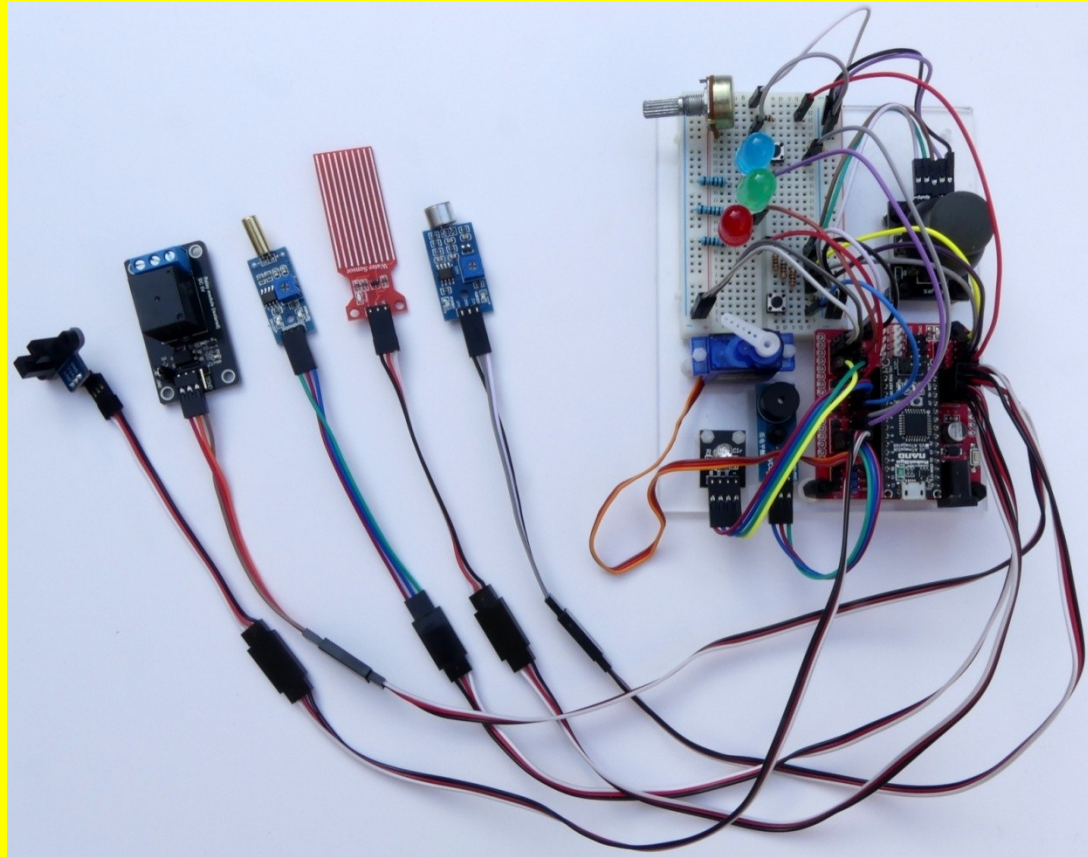
... am Ende der  
Einsteigerkurs



# ... am Ende der Zwischenstufe ...



# ... und am Ende des fortgeschrittenen Niveaus



# Teil 3: Laden der Skizze in den Arduino (wahlweise im Workshop, notwendig zuhause)

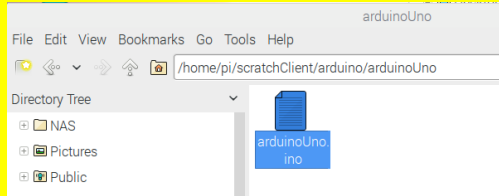
# Vorbereitung für die Programmierung des Arduino Nano



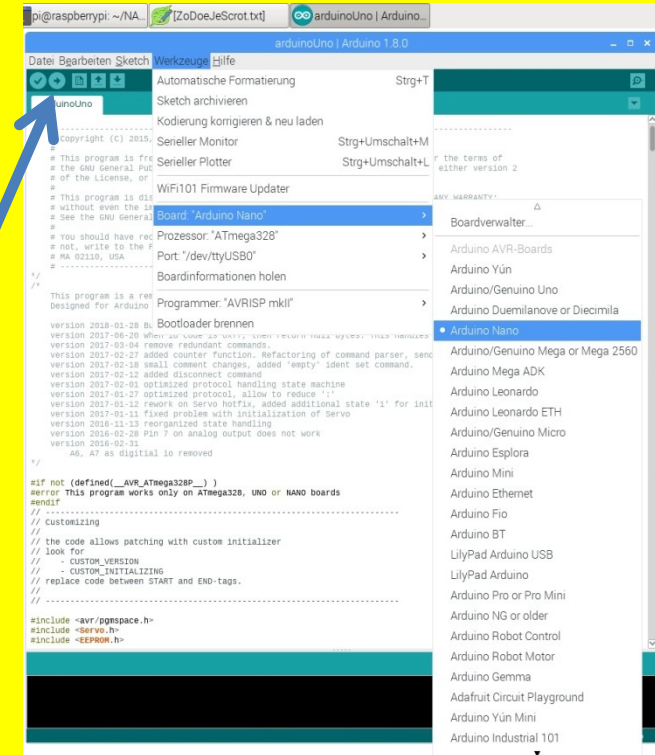
- Der Arduino muss ein Programm ausführen (in Arduino-Begriffen: Skizze), damit er mit dem Raspberry Pi kommunizieren und die Nachrichten von scratchClient verstehen kann.
- Wir müssen damit anfangen.
  - Dies ist jedoch bereits geschehen, sodass Sie die nächste Folie überspringen können, es sei denn, Sie möchten es selbst ausprobieren.
  - Zuhause müssen Sie das doch tun.

# Hochladen von scratchClient auf den Arduino

- Navigieren Sie zur ScratchClient-Skizze für Arduino Uno in `/home/pi/scratchClient/arduino/arduinoUno`



- Doppelklicken Sie, um die Arduino IDE zu öffnen
- Klicken Sie auf *Werkzeuge* und stellen Sie sicher, dass diese festgelegt sind:
  - Board: Arduino Nano
  - Prozessor: Atmega328
  - Port: Der Port, an dem der Arduino Nano angeschlossen ist (normalerweise `/dev/ttyUSB0`)
- Klicken Sie auf die Schaltfläche *Hochladen*.
- Warten Sie, bis der Upload abgeschlossen ist (ohne Fehler).

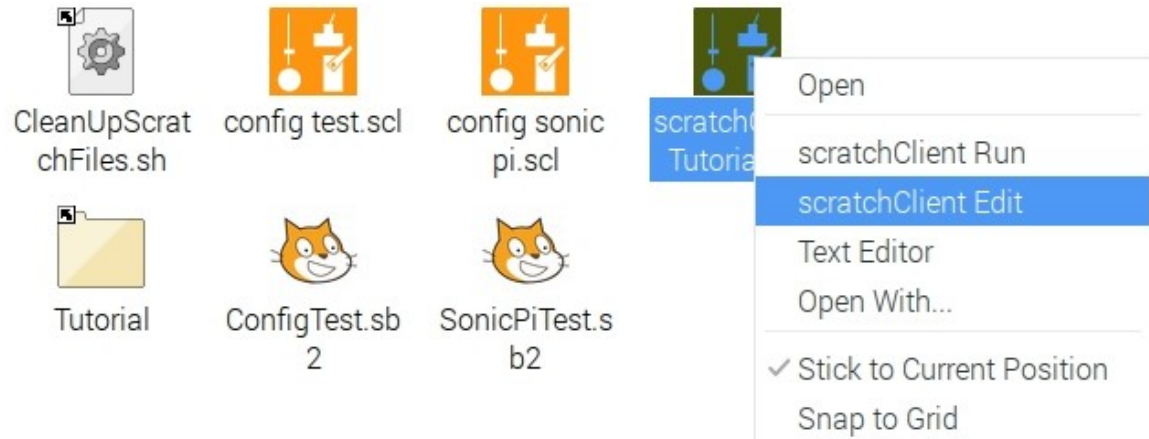


# Teil 4: Definieren der Konfiguration

Gib die Anschlüsse Namen und definiere den Zweck der  
Anschlüsse

# Starten des Konfigurationswerkzeugs

- Wir haben eine leere Konfigurationsdatei auf dem Desktop abgelegt:
  - *ScratchClient-Tutorial.scl*
- Klicken Sie mit der rechten Maustaste auf die Datei und wählen Sie *scratchClient Edit*





# Definieren Sie die erste Konfigurationsdatei

File Help

name: scratchCI Tutorial

name	arduino	direction	function	scratchName
D0		void		
D1		void		
D2		void		
D3		void		
D4		out	output	BigRedLED
D5		void		
D6		void		
D7		in	input_pullup	Button
D8		void		
D9		void		
D10		void		
D11		void		
D12		void		
D13		void		
A0				
A1				
A2				
A3				
A4				
A5		void		
A6		void		
A7		void		

D4  
Direction: out  
Function: output  
scratchName: BigRedLED

D7  
Direction: in  
Function: input\_pullup  
scratchName: Button

Parameter serial.device: /dev/ttyUSB0

Parameter serial.device: /dev/ttyUSB0

Parameter ident.check: ☒

Parameter ident.pattern:

type id message

INFO empty ident.pattern connects only to arduino with empty ident

```
<!-- id = 'D7' direction = 'in' function = 'input_pullup' -->
<output_value name="outputD7">
  <sensor name="Button"/>
</output_value>

<extension>
  <io dir="out" id="D4"/>
  <io dir="in" id="D7" pullup="on"/>
</extension>

<parameter name="serial.device" value="/dev/ttyUSB0"/>
<parameter name="serial.baud" value="115200"/>

<!-- optional parameters for IDENT check -->
<parameter name="ident.check" value="true"/>
<parameter name="ident.pattern" value=""/>
```

- **Doppelklicken** Sie auf eine Zelle, um ein Dropdown-Menü zu erhalten
  - Zuerst für die *Richtung*, dann für die *Funktion*.
- Achte darauf, dass Sie alle Anlüsse Namen geben, wenn Sie etwas anderes wählen als *void* in *direction*.
  - Stellen Sie also sicher, dass Sie nicht speichern, wenn Sie immer noch rote Ränder um Zellen haben. scratchClient kann mit einer solchen Konfigurationsdatei nicht starten.
- Das Tool überprüft falsche Konfigurationen.  
Beispiele:
  - Die Anlüsse 0, 1 und 13 können überhaupt nicht verwendet werden
  - Analog ist nur auf A0 bis A7 verfügbar
  - Die Anlüsse A6 und A7 können nur für analoge Eingänge verwendet werden
  - Anlüsse 9 und 10 können nicht für PWM verwendet werden, wenn ein Ansluß für Servo konfiguriert ist (siehe später)

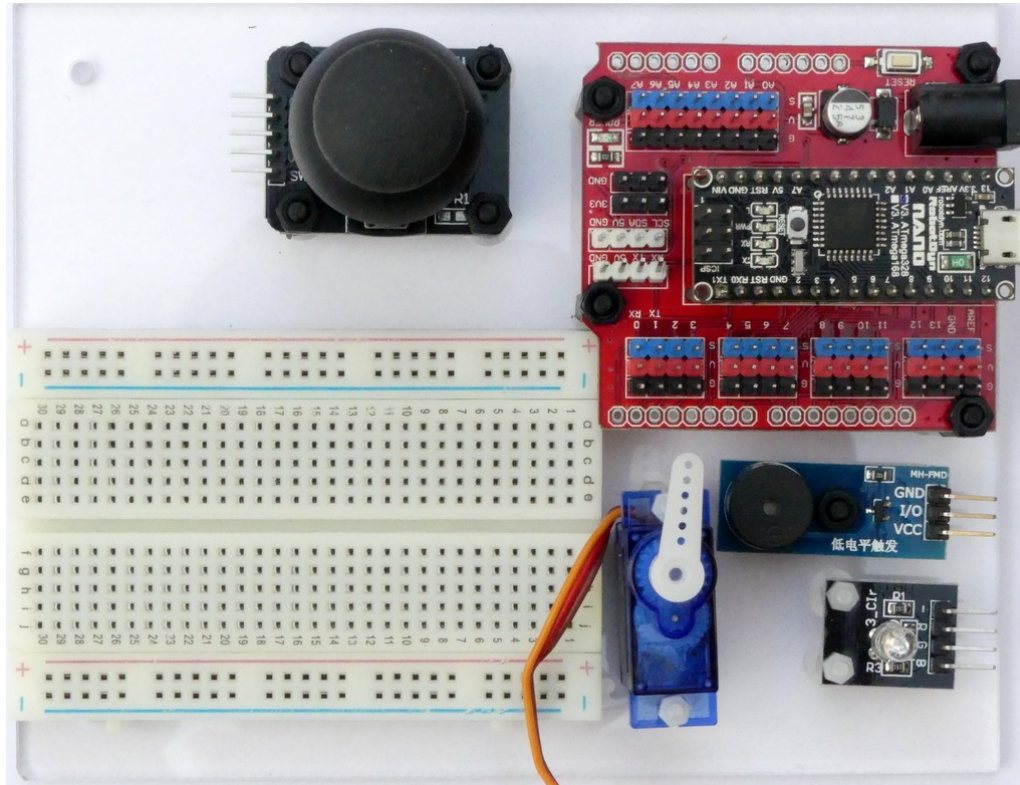


# Speichern Sie die Konfigurationsdatei

- Speichern Sie die Datei (Stgr-S oder File → Save).
- **Warnung:** Schließen der Datei ohne speichern verlieren Sie alle Änderungen.
- Lassen Sie das Werkzeug für die nächsten Übungen geöffnet.

# Teil 5: Verdrahten Sie die Platine und starten Sie das erste Setup

# Stellen Sie das Brett auf diese Weise vor sich hin



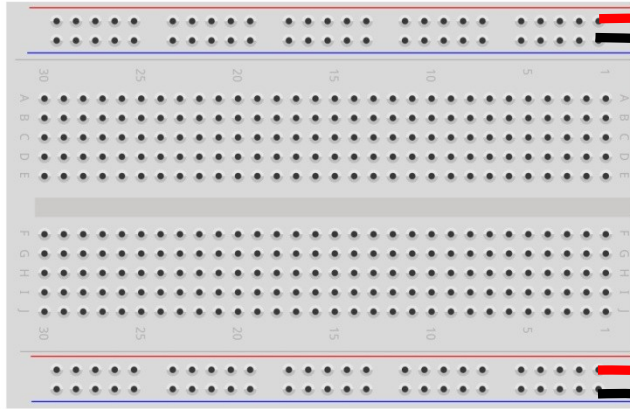
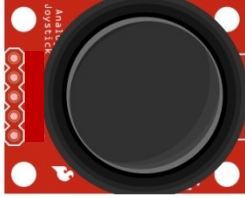
# Verwenden Sie kurze Drähte und verwenden Sie die angegebenen Löcher

- Es gibt einige kurze Drähte (10 cm) und einige längere (15 cm)
- Verwenden Sie den kürzesten, den Sie können
  - Holen Sie sich ein weniger chaotisches Setup
  - Sie könnten sonst später keine langen Drähte mehr haben
- Ignorieren Sie die Drahtfarben.
- Sie können jedoch prinzipiell an verschiedenen Stellen auf dem Steckbrett aufbauen ...

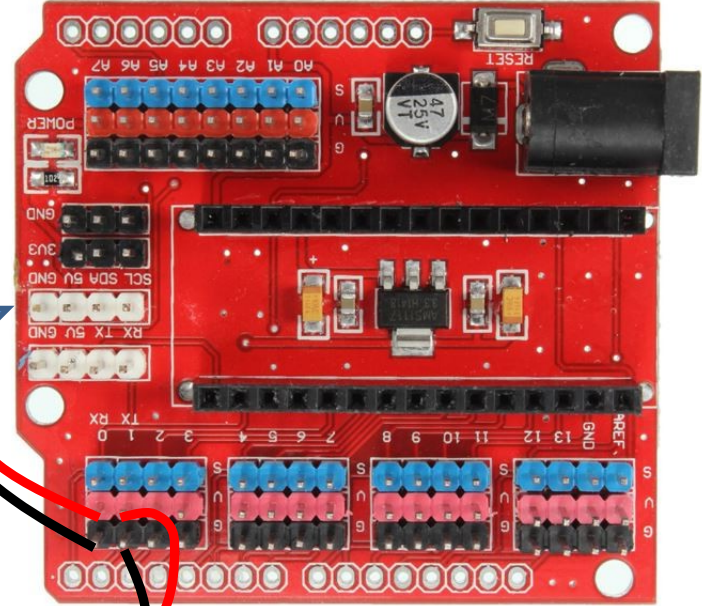
... bitte benutzen Sie die angegebenen Spalten, um zu vermeiden, dass im späteren Teil des Workshops der Platz auf dem Steckbrett knapp wird.

# Schließen Sie die Stromkabel an

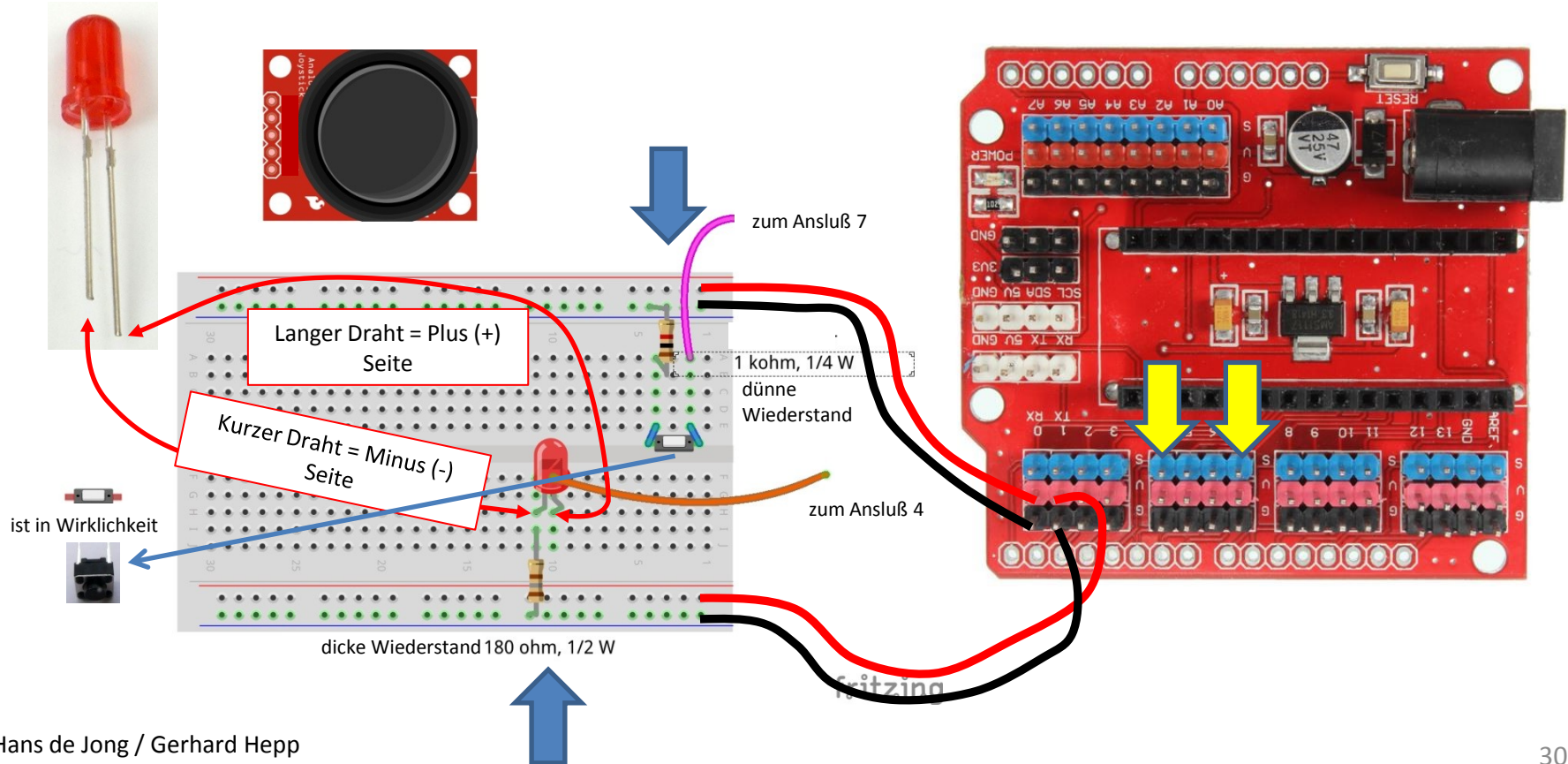
GND  
+5V  
VRx  
VRy  
SW



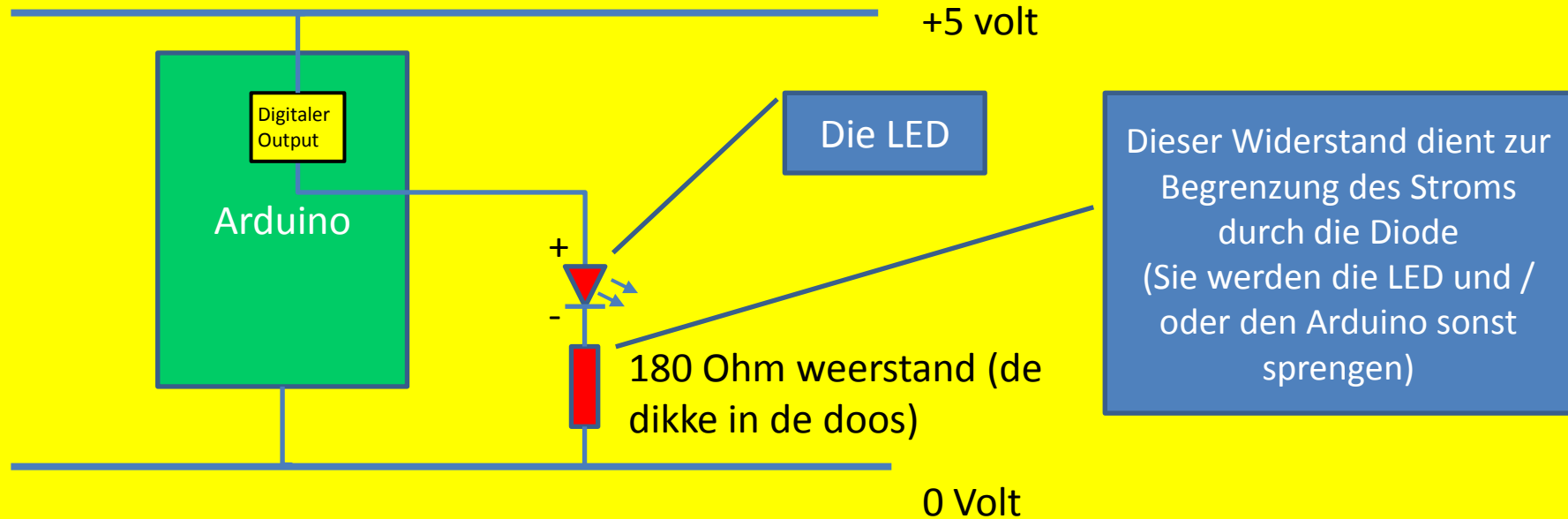
fritzing



# Setzen Sie die rote LED und den Knopf ein



# Warum setzt man einen Widerstand in Reihe mit der LED?



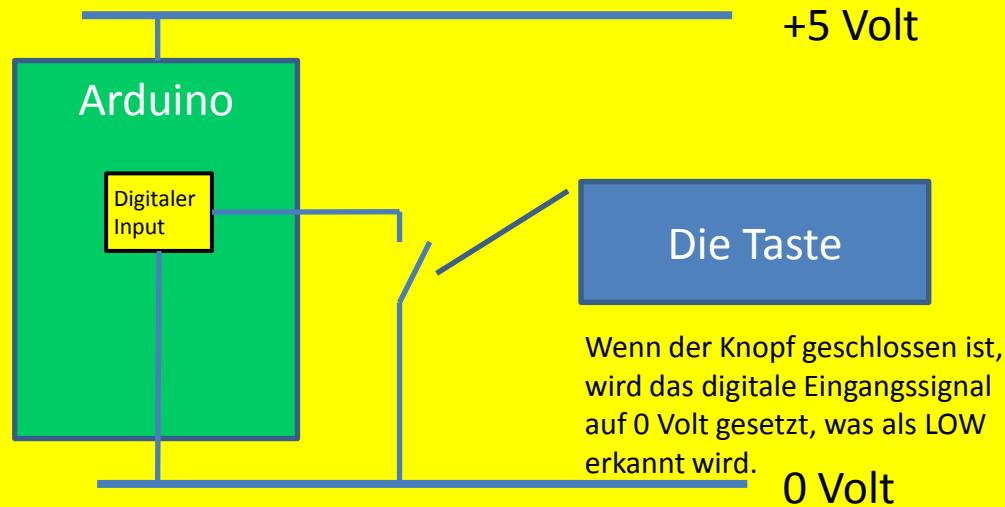
# Was benötigt Arduino an einem digitalen Eingangspin?



- Ein digitaler Eingangspin von Arduino muss entweder erhalten
  - 0-Volt-Eingang (tatsächlich gilt 0 bis 1,5 Volt als LOW-Eingangssignal)
  - 5-Volt-Eingang (tatsächlich werden 2,5 bis 5 Volt als HIGH-Eingangssignal betrachtet)
- Wenn es etwas zwischen 1,5 Volt und 2,5 Volt bekommt, wird die Interpretation nicht stabil sein (könnte LOW oder HIGH sein).
- Wenn kein Signal empfangen wird, ist die Interpretation des Signals nicht stabil (könnte LOW oder HIGH sein).

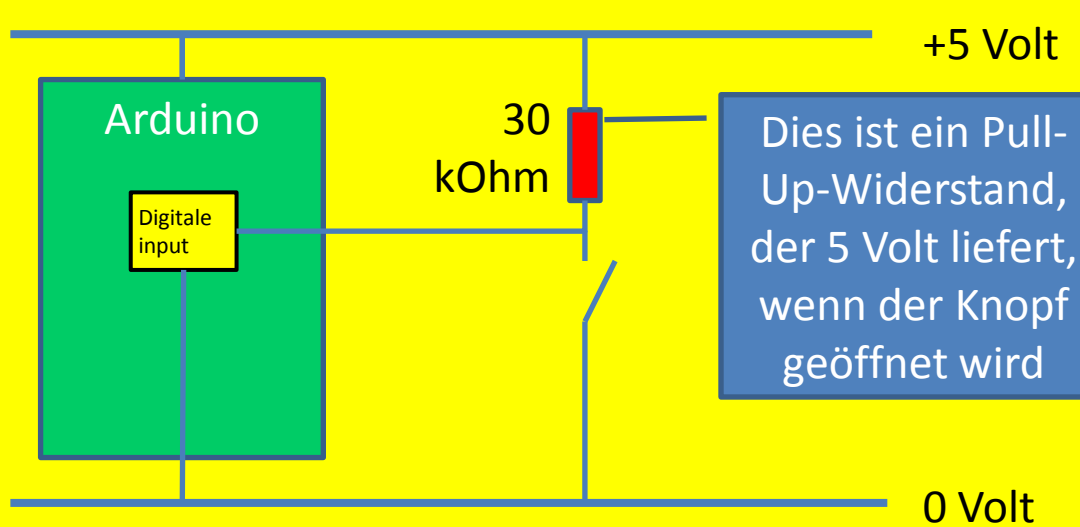


# Einen LOW-Wert erzeugen



Aber was wird der Arduino feststellen,  
wenn die Taste geöffnet ist?

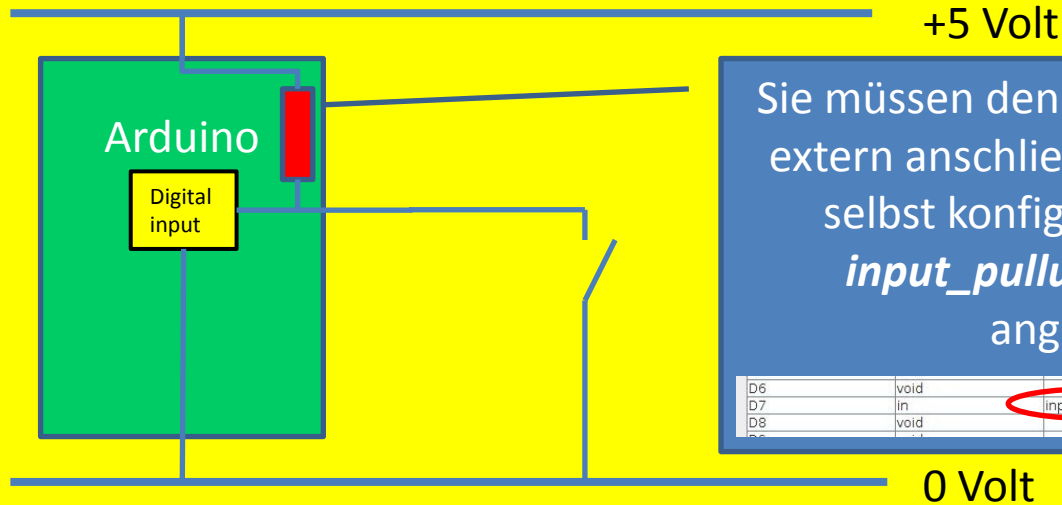
# Einen HOHEN Wert erzeugen



Wenn die Taste gedrückt wird, verbindet es den Eingang des Arduino mit 0 Volt. Es wird ein kleiner Strom durch den Widerstand fließen, aber der Eingang wird 0 Volt sein.

Wenn der Schalter geöffnet ist, zieht der Widerstand den Eingang auf 5 Volt hoch. Ein sehr kleiner Strom fließt durch den Widerstand, ist aber klein genug, dass die Spannung am digitalen Eingang sehr nahe bei 5 Volt = HOCH liegt.

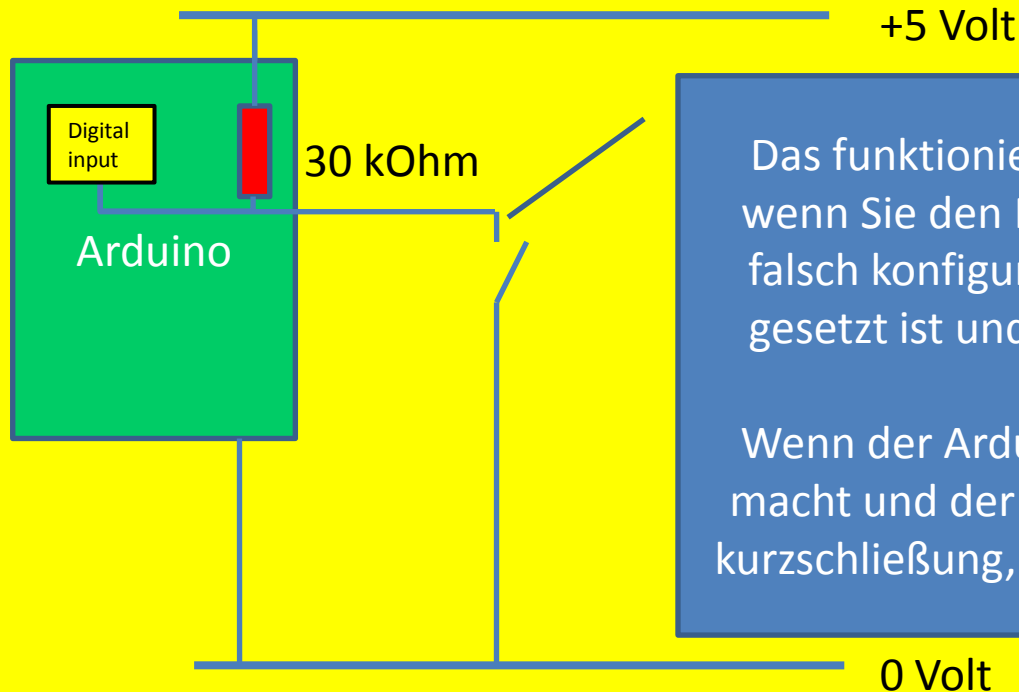
# Erzeugen eines HIGH-Wertes ohne externen Widerstand



Sie müssen den Widerstand jedoch nicht extern anschließen. Es kann im Arduino selbst konfiguriert werden, indem ***input\_pullup*** anstelle von ***input*** angegeben wird.

D6	void		
D7	in		
D8	void	<b>input_pullup</b>	Button
D9	void		

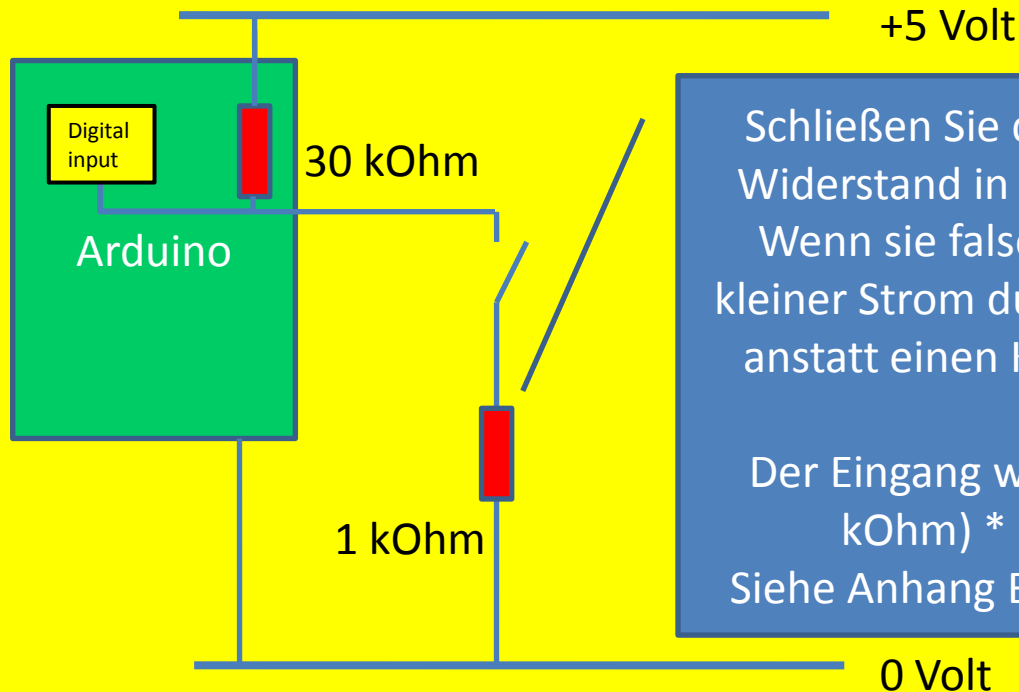
# Gefahr bei Fehlkonfiguration



Das funktioniert gut, aber was passiert, wenn Sie den Pin für den Digitalausgang falsch konfigurieren, der Wert auf HIGH gesetzt ist und die Taste gedrückt wird?

Wenn der Arduino den Pin HIGH (5 Volt) macht und der Taster mit LOW (0 Volt) → kurzschließung, wird der Arduino springen.

# Widerstand in Reihe, um Schäden bei Fehlkonfiguration zu vermeiden



Schließen Sie daher immer einen 1 kOhm Widerstand in Reihe mit dem Schalter ein. Wenn sie falsch konfiguriert ist, wird ein kleiner Strom durch den Widerstand fließen, anstatt einen Kurzschluss zu verursachen.

Der Eingang wird  $1 \text{ kOhm} / (30 \text{ kOhm} + 1 \text{ kOhm}) * 5 \text{ Volt} = 0,15 \text{ Volt}$  sein. Siehe Anhang B für weitere Informationen.

# Überprüfen / überprüfen

- Bitte überprüfen Sie jetzt **beide von Ihnen**, dass die Verkabelung korrekt ist.

# Dinge zusammenbringen

- Verbinden Sie den 9-Volt-Stecker mit der Platine
  - Obwohl solange Sie nur einen LED benutzen, reicht die USB-Anschluß
- Verbinden Sie den USB-Anschluß mit dem Arduino
- Doppelklicken Sie *scratchClient-Tutorial.scl* auf dem Büroblatt, um scratchClient mit der gerade aktualisierten Konfigurationsdatei zu starten



scratchClient-  
Tutorial.scl

- Es wird auch ein Browserfenster geöffnet, in dem Sie Variablen sehen können
  - Später in der Präsentation erläutert
- Im Terminalfenster sehen Sie Beschwerden, dass scratchClient keine Verbindung zu Scratch hat
  - Was logisch ist, weil Scratch noch nicht gestartet ist.

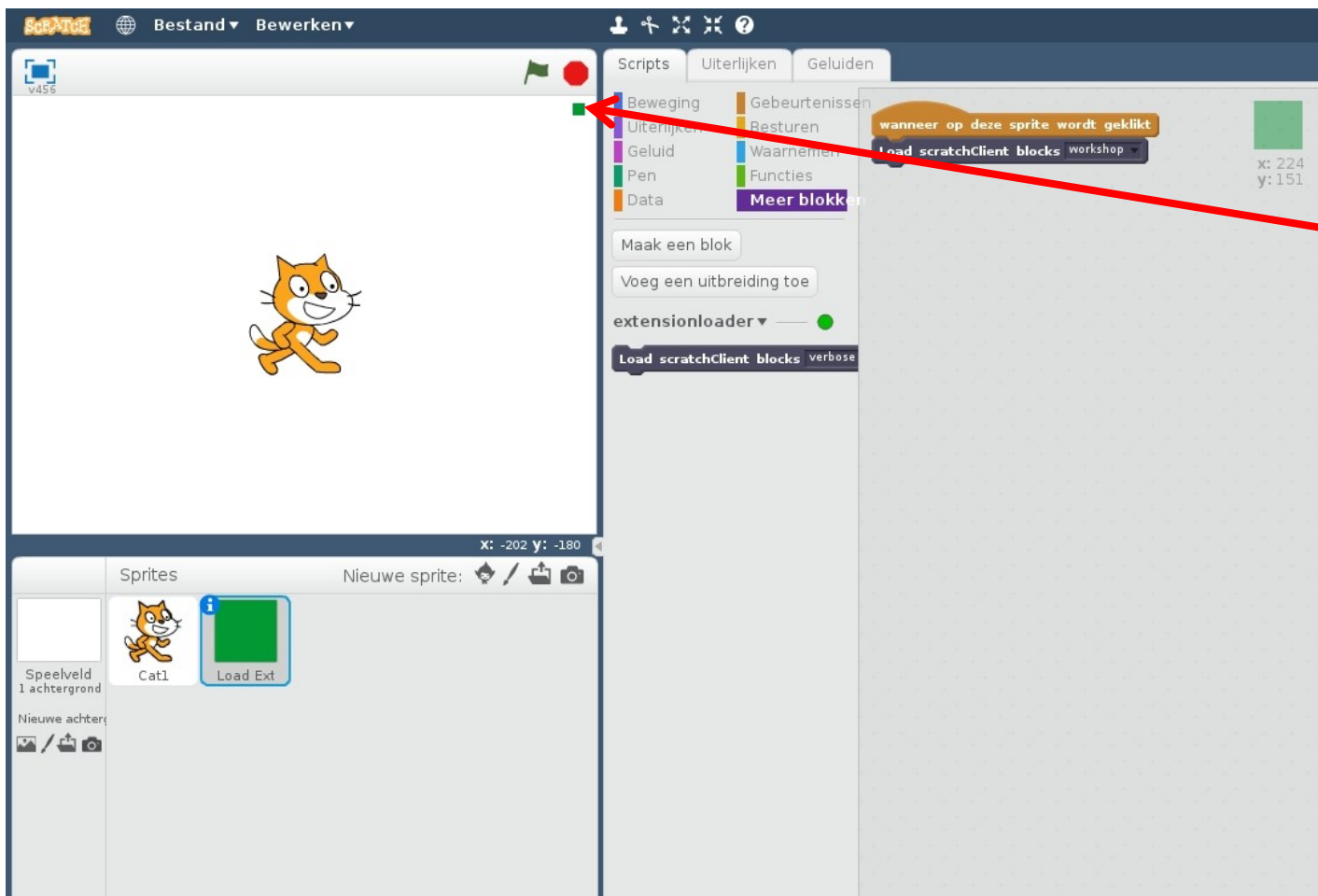
# Start Scratch 2

- Der einfachste Weg ist, die Datei auf dem Büroblatt zu starten
- Doppelklicken Sie auf das Symbol

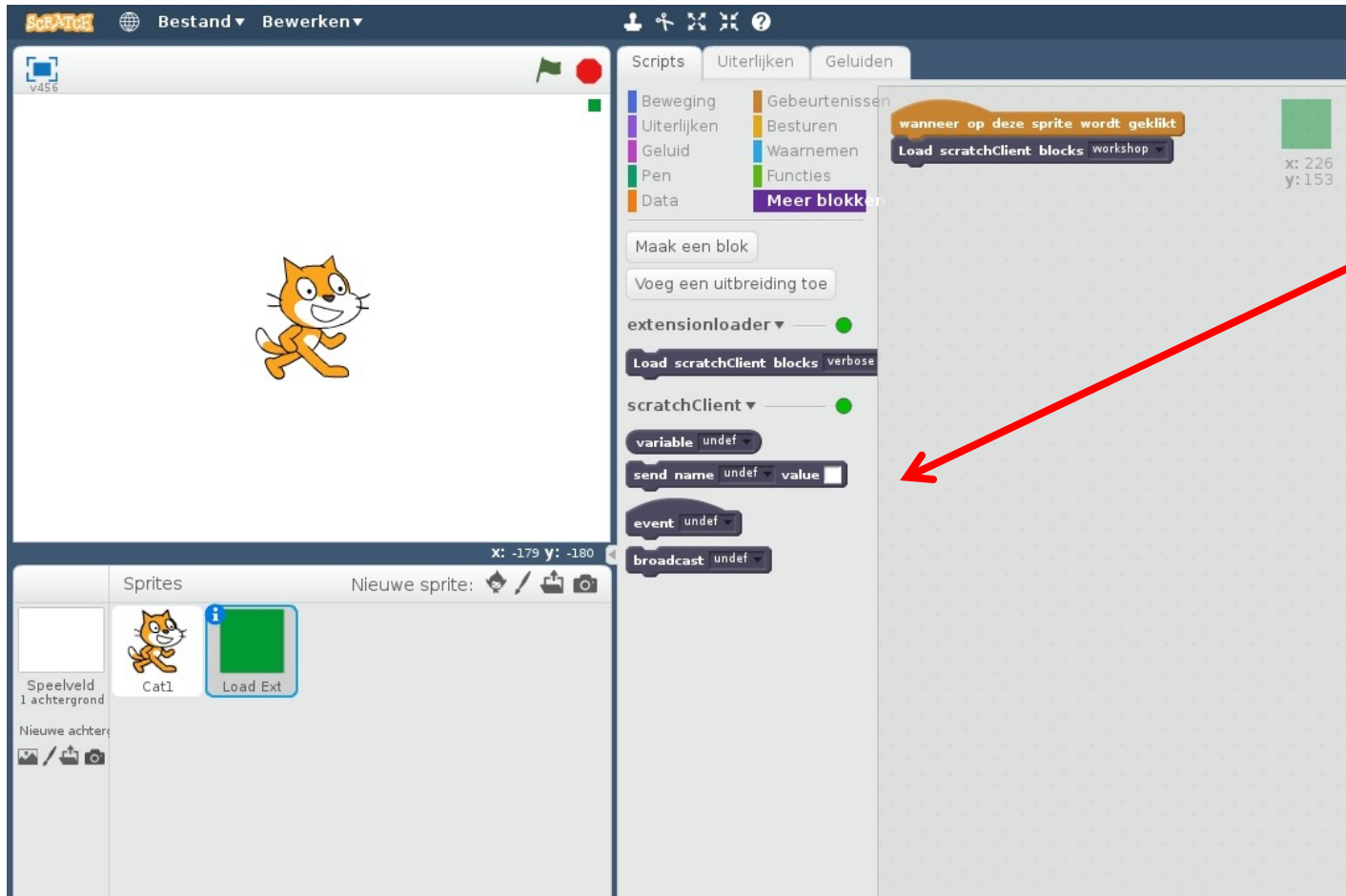


- Sehen Sie im Anhang, wie Sie eine solche Datei selbst erstellen können
  - In der aktuellen Version von Scratch 2 gibt es ein paar Eigenheiten





Klicken Sie auf dieses Sprite, um die Erweiterungsblöcke sichtbar zu machen.



Durch Klicken auf das grüne Sprite wurden diese Erweiterungsblöcke angezeigt.

# Erstellen Sie das Scratch-Programm

- Erstellen Sie dieses Programm in Scratch und probieren Sie es aus (Legen Sie es auf den Bildschirm oder das Cat-Sprite)
- Versuchen Sie es (klicken Sie auf die Blöcke oder klicken Sie auf die grüne Flagge über dem Animationsfenster).
- Digitaleingang (Taste):
  - 0 = gedrückt
  - 1 = nicht gedrückt
- Digitalausgang (LED)
  - 0 = aus
  - 1 = ein
- Analysieren Sie, wie das Programm funktioniert.



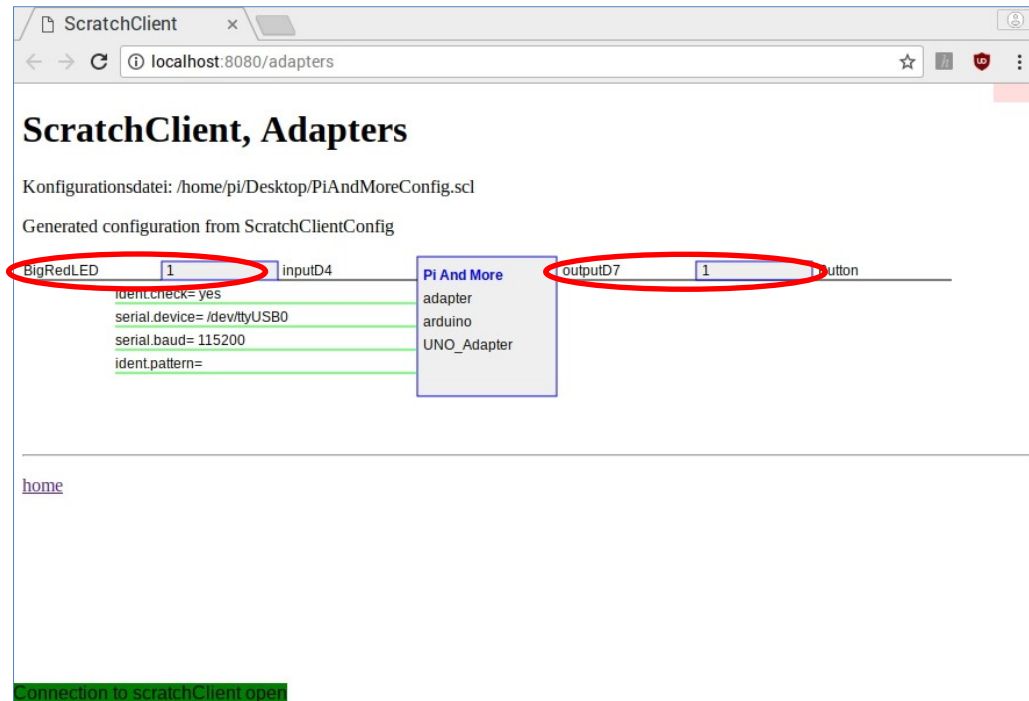
Verwenden Sie diese Pfeile, um den richtigen Wert auszuwählen (das Schwarz auf einem grauen Hintergrund macht es nicht sehr sichtbar)

# Funktioniert es?

(siehe nächste Folie für Hilfe)

- Wenn die LED am Arduino langsam blinkt, **nur dann** ist die Konfiguration heruntergeladen und **nur dann** funktioniert scratchClient.
  - Es kann 10 Sekunden dauern, bis sowohl scratchClient als auch die ScratchClient-Erweiterungen in Scratch 2 geladen sind, bevor dies geschieht.
  - Also 10 Sekunden nachdem Sie auf das grüne Sprite geklickt haben.

# Sie können die Werte überwachen, die ausgetauscht werden

A screenshot of a web browser window showing the ScratchClient interface. The browser's address bar displays "localhost:8080/adapters". The page title is "ScratchClient, Adapters". Below the title, it says "Konfigurationsdatei: /home/pi/Desktop/PiAndMoreConfig.scl" and "Generated configuration from ScratchClientConfig". The main content area shows a configuration for a "Pi And More" adapter. On the left, under "BigRedLED", there is a variable "1" in a text box, which is circled in red. This is connected to "inputD4". In the center is a box labeled "Pi And More" with sub-labels "adapter", "arduino", and "UNO\_Adapter". On the right, there is a variable "1" in a text box next to "outputD7", also circled in red, which is connected to a "button". Below the configuration, there is a "home" link and a green status bar at the bottom that says "connection to scratchClient open".

- Zusammen mit dem scratchClient wird der Browser mit der URL localhost: 8080 / adapter geöffnet
- Beachten Sie die Eingabe- und Ausgabeanweisungen:
  - Die Ausgabe eines Adapters ist eine Eingabe für Scratch
  - Die Ausgabe von Scratch ist eine Eingabe für die Adapter.
  - Daher scheinen die Eingabe und Ausgabe der Namen umgekehrt zu sein, was in der Konfigurationsdatei enthalten ist.
  - Daher beziehen Sie sich am besten auf die Variablennamen.
- Sie werden sehen, dass Werte nur angezeigt werden, nachdem sie sich geändert haben (andernfalls wird ein Fragezeichen (?) Anzeigt).
- Nicht für den Moment, aber Sie können auf das Feld klicken und einen Wert eingeben, der dann in die entsprechende Richtung gesendet wird.

# Ändern Sie das Programm

- Nehmen Sie eine Änderung am Programm vor, so dass die LED beim Drücken der Taste leuchtet.
- Es gibt (mindestens) zwei Möglichkeiten
- Sie benötigen die hier abgebildeten Blöcke



Blöcke für Methode 1





Blöcke für Methode 2

# Was ist, wenn es nicht funktioniert?

- Überprüfen Sie, ob mehrere Scratch-Instanzen geöffnet sind
  - scratchClient kann immer nur mit einem Scratch arbeiten (unabhängig davon, ob es sich um Scratch 1.4 oder Scratch 2 handelt).
- Überprüfen Sie, ob die blaue LED am Arduino Nano *langsam* blinkt.
- Manchmal, besonders nach dem Neustart, wenn alles in Ordnung scheint, kann das Trennen und erneut Verbinden des Arduino Nanos helfen.
- Versuchen Sie, die Variablen zu überwachen, siehe vorher.

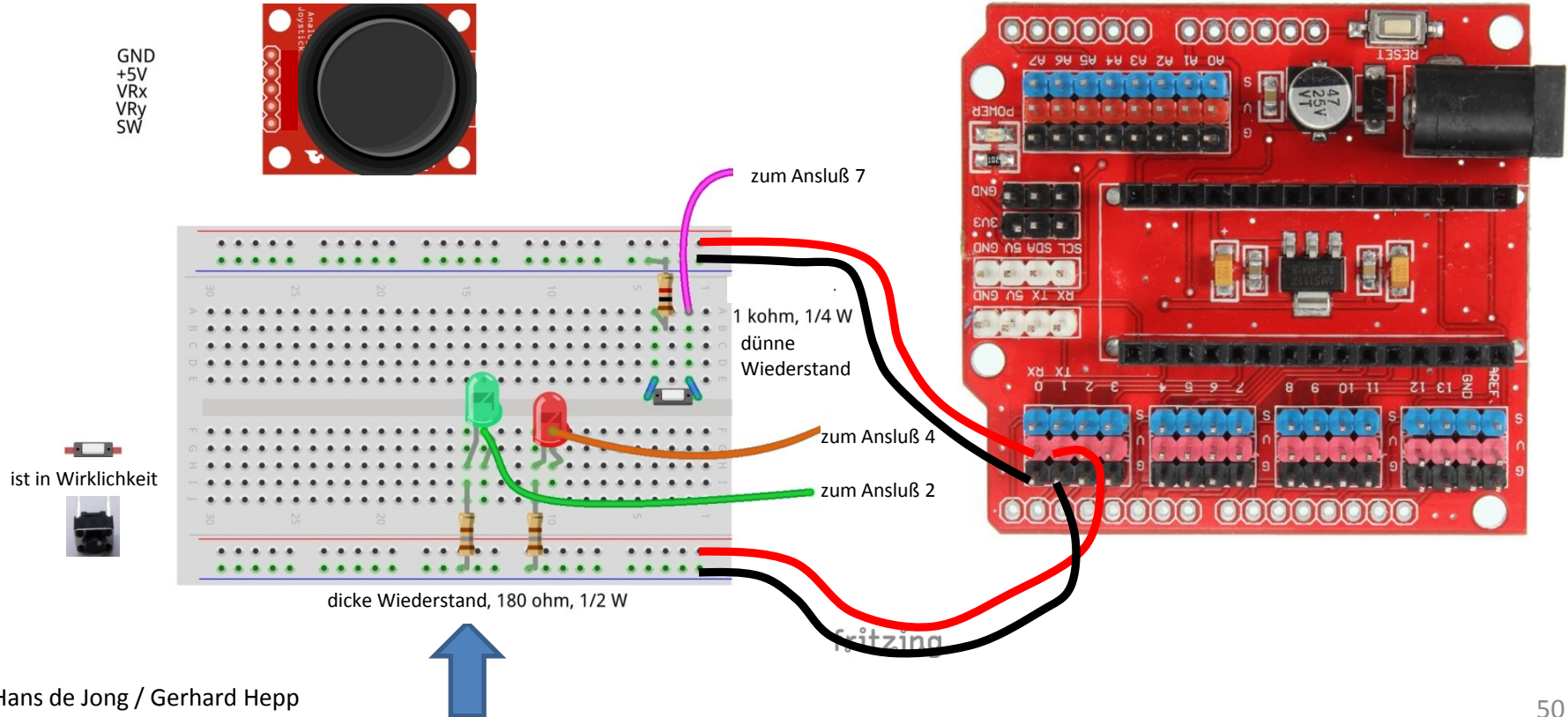
# Eine Erinnerung ...

- Speichern Sie Ihr Scratch-Programm regelmäßig. Andernfalls wird es beim Ausschalten verloren gehen.
  - Ein Stromausfall kann leicht passieren, da Sie Kabel ziehen und den Stromanschluss des Raspberry Pi beeinträchtigen können.
- Hier kommen einige aktuelle Scratch 2 Anomalien auf RPi
  - Sie sollten keine Leerzeichen in Dateinamen verwenden (wenn Sie dies tun, werden sie entfernt)
  - Scratch 2 vergisst den Ordner, in dem die Datei von (Büroblatt) geöffnet wurde. Speichern Sie es einfach in `/home/pi`, wie Scratch vorschlägt
  - Scratch 2 speichert es mit einer Erweiterung `.sbx` unabhängig von der Angabe.
- Also einmal nach dem Einschalten
  - Doppelklicken Sie auf dem Desktop auf `CleanUpScratchFiles.sh` einmalig aber ersten Mal dass RPi gestartet ist.  `CleanUpScratchFiles.sh`
  - Es macht ein vorschlag om automatisch vom crontab zu starten. Empfohlen. Das ist so im Workshop.
  - Dies überwacht permanent den Ordner `/home/pi`, und
    - Verschiebt alle Dateien mit `.sbx` auf den Desktop als `.sb2`-Dateien
    - Alle Leerzeichen in Dateinamen werden entfernt
    - Wenn sich bereits eine solche Datei auf dem Desktop befindet, wird sie in den Ordner `ScratchProgramsBackup` auf dem Desktop verschoben  `ScratchProgramsBackup`
- Es wird auch eine Warnung ausgegeben, wenn Sie die Datei eine Zeitlang nicht gespeichert haben, wenn zwei Scratch-Instanzen geöffnet sind, wenn scratchClient nicht ausgeführt wird und mehr.
- Wenn Sie eine gespeicherte Datei erneut öffnen, klicken Sie erneut auf das grüne Sprite, um die Erweiterungsblöcke zu laden



# Teil 6: Hinzufügen der großen grünen LED

# Hinzufügen der grünen LED



# Überprüfen und erneut verbinden

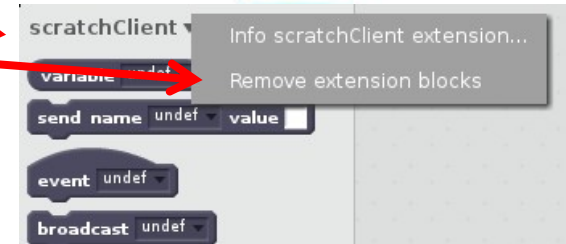
- Überprüfen Sie die korrekte Verkabelung
- Schalten Sie die 9V-Stromversorgung ein
- Schließen Sie das USB-Kabel erneut an

# Aktualisieren Sie die Konfigurationsdatei und starten Sie scratchClient neu


- Benutze das Config-Tool (welches noch geöffnet sein sollte)
- Definieren Sie einen Ausgang (direction: out, function: output) an Anluß 2 und nennen Sie ihn *BigGreenLED*
- Speichern Sie die Konfigurationsdatei (und lassen Sie das Werkzeug geöffnet)
- Doppelklicke *scratchClient-Tutorial.scl* erneut auf dem Desktop
- Dies wird die vorherige Instanz von scratchClient stoppen und mit der aktualisierten Konfiguration neu starten.
- Schmeissen Sie die scratchClient Erweiterung weg
  - Klicke hier rechts
  - Wähle Remove extension blocks
- Füge die Erweiterung nochmals hinzu (klicke auf das grüne Viereck)
  - Sie werden sehen in der *Weitere Blöcke* Abschnitt daß Sie wählen können um Werten nach BigGreenLED zu senden



scratchClient-Tutorial.scl



# Scratch aktualisieren

- Fügen Sie Code hinzu, der das tut:
  - Drücken Sie die Taste: Die LED leuchtet
  - Noch einmal drücken: LED erlischt
- Sie benötigen diese Programmelemente. 



# Dies ist das Ende des Anfängerniveau

- Sie haben Digital Out (LED) und Digital In (eine Taste) funktionieren
- Sie wissen, wie man mit scratchClient und Scratch 2 arbeitet
- Sie wissen, wie Sie scratchClient konfigurieren
- Wenn Sie wollen und Zeit haben
  - Sehen Sie sich einige der gelben Folien an, wenn Sie sie früher übersprungen haben
  - Machen Sie weiter mit einer der nächsten Ebenen
    - Diese befinden sich in verschiedenen Dateien
- Andernfalls,
  - Dateien auf Ihren USB-Stick kopieren und aufräumen, siehe weiter

# Übersicht der nächsten Niveaus

- Zwischenniveau (Material nur auf Englisch)
  - Analogeingang: Potentiometer
  - Pulsweitenmodulation (wegen Abwesenheit eines analogen Ausgangs)
    - Dimmen einer LED
    - Ein Servo steuern
    - Einen Summer steuern
- Fortgeschrittenes Niveau (Material nur auf Englisch)
- Expertenniveau (Material nur auf Englisch)

# Teil 10: Nimm Ihre Arbeit mit nach Hause



# Wollen Sie Ihre Arbeit nach Hause mitnehmen?

- Wenn Sie Ihren eigenen USB-Stick mitgebracht haben, schließen Sie ihn an und kopieren Sie die Datei *scratchClient-Tutorial.scl* auf den Büroblatt sowie alle auf dem Büroblatt erstellten .sb2-Dateien
- Der Rest des Materials, können Sie von *www.github.com* herunterladen, suchen Sie nach *scratchClient-Tutorial*
- Nehmen Sie die Karte mit, um sich zu merken, wo das Material auf GitHub zu finden ist.

# Teil 11: Zusammenfassung & Mitnehmen

# Takeaways des Anfänger-Workshops

- Mit scratchClient definieren Sie:
  - Funktion jedes Anschluß
  - Symbolischer Name für jeden konfigurierten Anschluß
- scratchClientConfig ist das Tool zum Einrichten der Konfiguration
- Starten Sie scratchClient neu, nachdem Sie die Konfiguration geändert haben
- Setzen Sie einen Widerstand in Reihe mit LEDs
- Setzen Sie einen Widerstand in Reihe mit Schaltern
- Konfigurieren Sie einen Pull-up-Widerstand, wenn das Eingangssignal zwischen 0 Volt und offen liegt und nicht zwischen 0 Volt und 3 bis 5 Volt.
- In Scratch 2 verwenden Sie den Erweiterungsblock für scratchClient, um Blöcke zu erhalten, die Sie für die Interaktion mit scratchClient verwenden können
- Sie können den Wert aller Anschlüsse vom Browser aus überwachen
- **scratchClient kann viel mehr ...**
- Funktionen, die einen Anschluß auf Arduino haben kann:
  - Digitaleingang
  - Digitalausgang
  - Analogereingang \*
  - *Kein analoger Ausgang*
  - Pulsweitenmodulation als Alternative \*
    - Zum Modulieren der Helligkeit einer LED
    - Zum Steuern eines Servos
    - Zum Steuern eines Summers
  - Es gibt einige mehr, siehe den fortgeschrittenen Workshop
  - Sie können Pull-up-Widerstände an Digitaleingang konfigurieren

\* Siehe Zwischenniveau

# Teil 12: Aufräumen / Abbau

# Wenn Ihr scratchClient-Tag hier endet ...

- Trennen Sie die Platine von USB und von die 9V Stromversorgung
- Bitte entfernen Sie alle Komponenten und Kabel vom **Steckbrett**
- Entfernen Sie alle Drähte von der **Arduino-Platine**.
- **Lass die Drähte auf der 3-farbigen LED** (Sie haben das nicht benutzt)
- **Lassen Sie die Kabel am Summer**
- Wenn etwas kaputt ist, bitte
  - Werfen Sie es weg oder reichen Sie es ein (wenn es unklar ist)
  - Schreiben Sie eine Notiz im Schachtel, dass es fehlt
  - Legen Sie nichts was kaputt ist zurück in die Schachtel
- Raspberry Pi absließen
- Lassen Sie uns wissen, was Sie über diesen Workshop gedacht haben, jetzt mündlich oder später per E-Mail
  - [hans.piam@hanselma.nl](mailto:hans.piam@hanselma.nl)

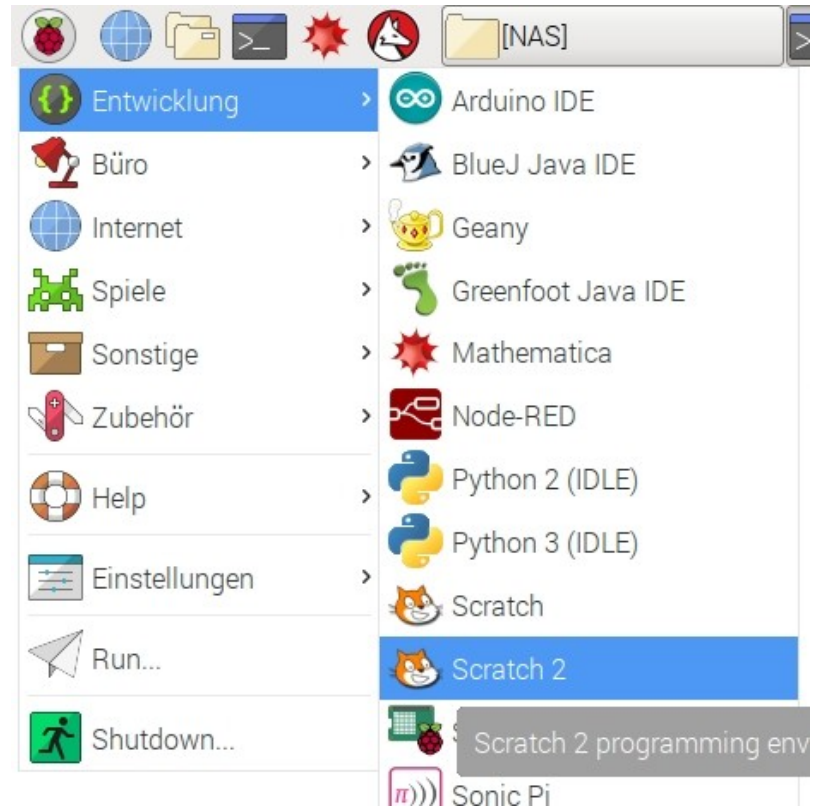
# Anhang A

## Erweiterungsblöcke in Scratch 2

# Wie füge ich Erweiterungsblöcke hinzu?

- Stap 1: Zorg dat scratchClient is loopt.
- Aufgrund einer Reihe von Besonderheiten ist das Laden von Erweiterungsblöcken etwas umständlich.
- Sie können die zuvor erläuterte Methode verwenden (klicken Sie auf das grüne Punktsprite).
- In diesem Anhang wird erläutert, wie Sie dies von Scratch aus tun.  
Schritt 1: Stellen Sie sicher, dass der scratchClient ausgeführt wird.

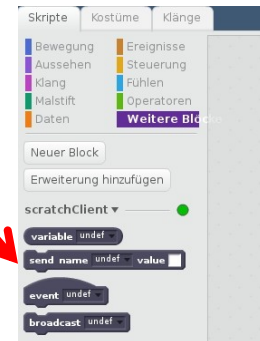
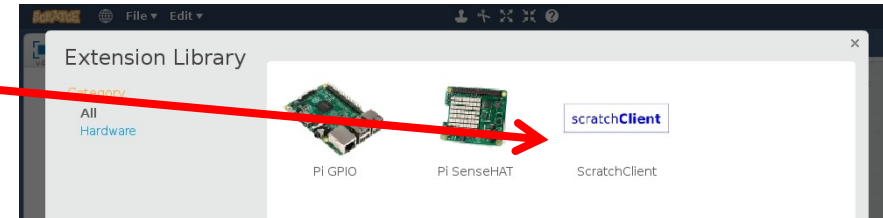
# Start Scratch 2





# Starten Sie Scratch 2 und erhalten SiescratchClient Blöcke

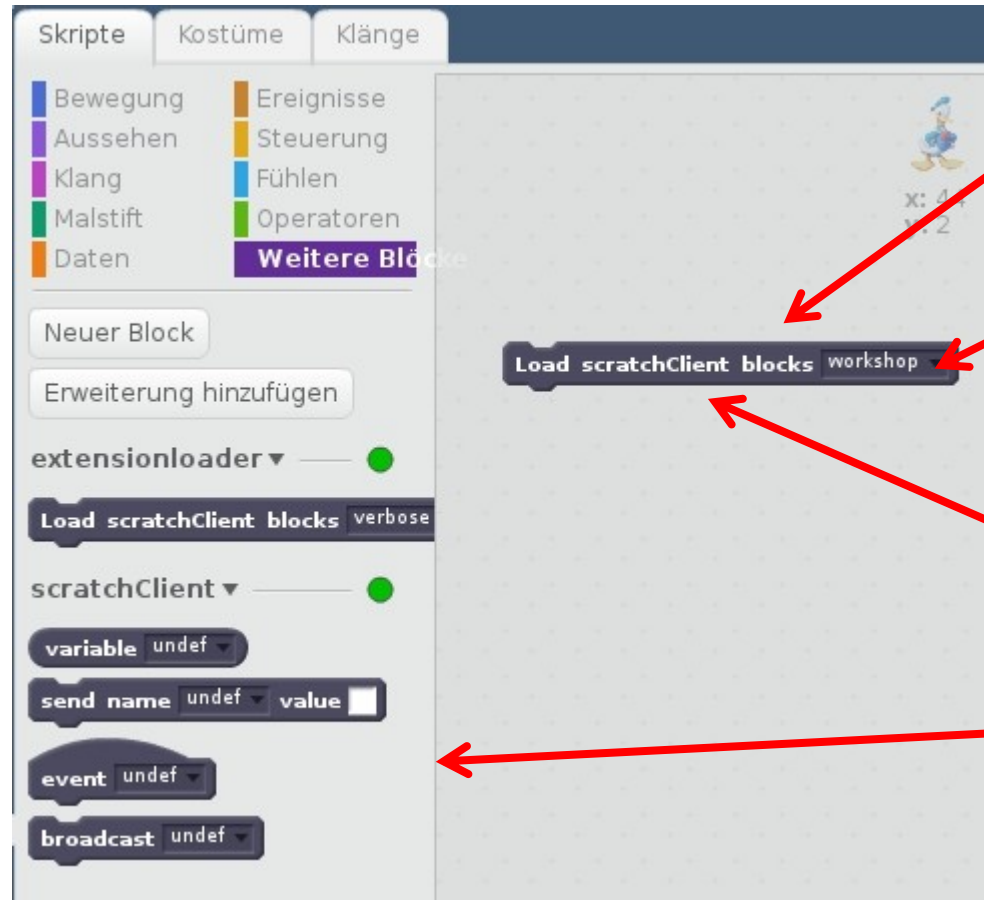
- Klicken Sie auf *Weitere Blöcke*
- Klicken Sie auf *Erweiterung hinzufügen*
- Wählen Sie scratchClient
- Jetzt werden die zusätzlichen Blöcke von scratchClient eingeschlossen.
- Warten Sie 10 Sekunden und schauen Sie, ob die blaue LED am Arduino Nano langsam statt schnell blinkt. Ist dies der Fall, ist die Verbindung hergestellt.



# Schritt 4-7: Laden scratchClient blocks



(2 von 2)



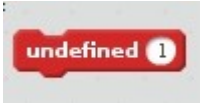
Schritt 4: Ziehen Sie diesen Block in den Bereich Scripts

Schritt 5: Hier *Workshop* auswählen

Schritt 6: Klicken Sie auf den Block, damit die zusätzlichen Blöcke angezeigt werden.

Schritt 7: Sehen Sie, wie die Blöcke erscheinen

# Die Blöcke werden nicht gespeichert

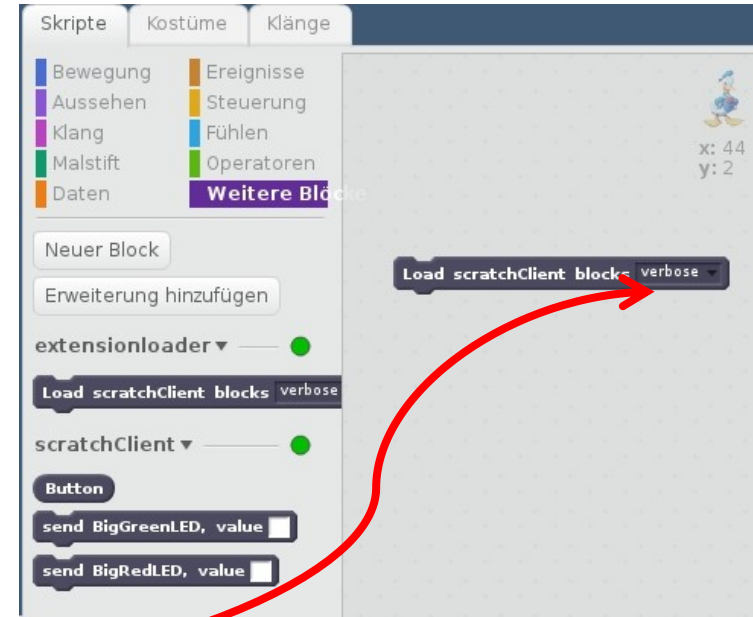
- Wenn Sie die Datei nach dem Speichern erneut öffnen, dann
  - Ist der scratchClient-Erweiterungslader noch vorhanden
  - Sind die echten Blöcke verschwunden
  - Und Sie sehen viele solcher Blöcke: 
- Daher müssen Sie nach dem erneuten Öffnen die Schritte 4 - 7 erneut wiederholen.
- Eine andere Methode finden Sie auf der nächsten Seite.

# Alternative für die Schritte 4 - 7

- Anstatt die Schritte 4 - 7 jedes Mal manuell durchzuführen, können Sie:
- Importieren Sie ein Figur  
*Load Name-Value Extension Blocks.sprite2* aus dem Ordner  
*~/scratchClient-Tutorials/scratchClientExtension/Sprites*
  - Denken Sie daran, dass beim Laden eines Sprites in Scratch 2 auf RPi die Maustasten manchmal tot sind und Sie mit den Pfeilen navigieren müssen (wählen Sie mit Return / Enter). Esc bringt Sie aus dieser Situation heraus.
- Dieses Sprite hat ein kleines grünes Quadrat, auf das Sie klicken können, um die zusätzlichen Blöcke jedes Mal zu laden, wenn Sie die Datei erneut öffnen.

# Eine andere Darstellung von Blöcken

- ScratchClient-Blöcke können auch mit einem Block pro Eingabe- oder Ausgabewert angezeigt werden.
- Sie erhalten dann z.B.
  - Ein separater Block für *BigRedLED*
  - Ein separater Block für *BigGreenLED*
  - Ein eigener Block für jeden Eingangsvariable (hier nur Button)
- Sie können diese Ansicht aufrufen, indem Sie im load Block die Option *verbose* auswählen



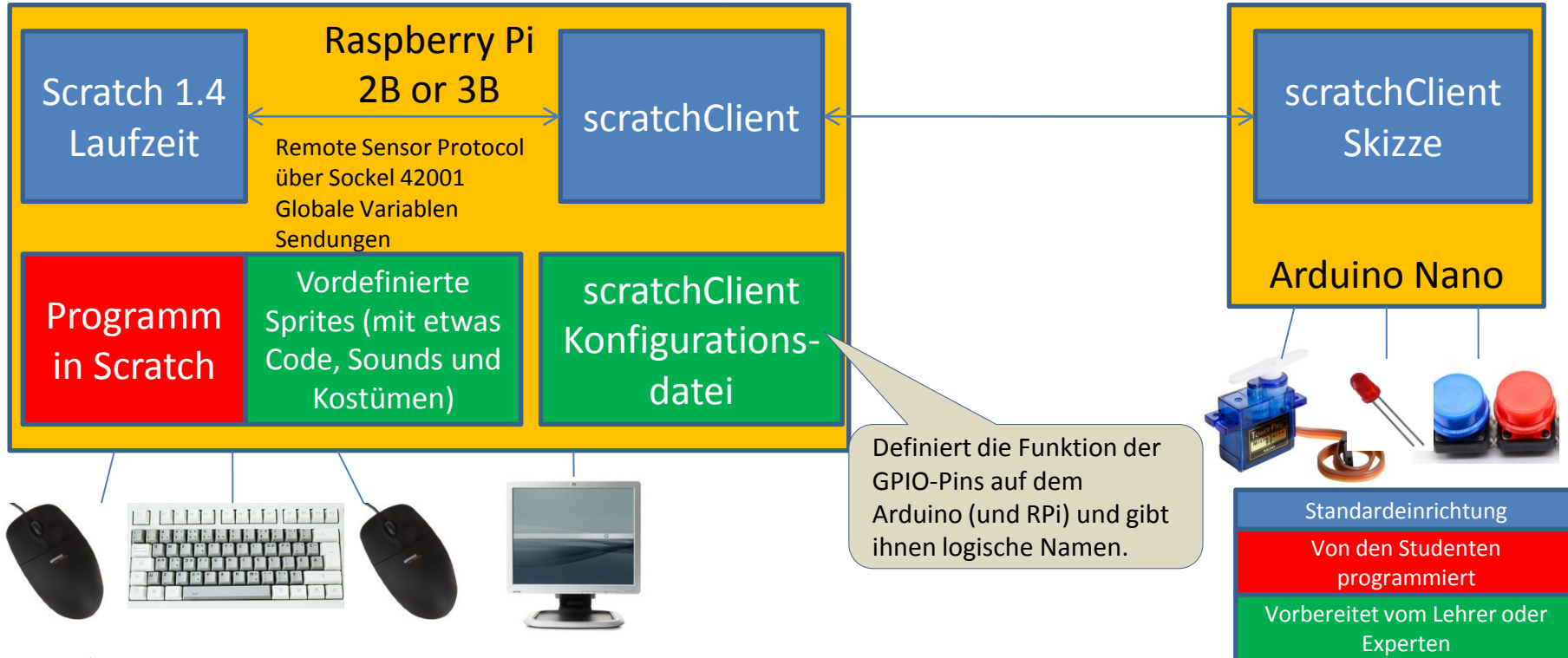
# Einfache Möglichkeit, die *verbose* Version der Blöcke zu erhalten

- Nehm die Figur *Load Verbose Extension Blocks.sprite2* von den Ordner *~/scratchClient-Tutorials/scratchClientExtension/Sprites*
  - Denken Sie daran, dass die Maustasten beim Laden möglicherweise tot sind, siehe oben.
- Dieses Sprite hat einen kleinen grünen Punkt, auf den Sie klicken können, um die zusätzlichen Blöcke jedes Mal zu erhalten, wenn Sie die Datei erneut öffnen.
  - Vergleichen Sie das mit dem Laden der Name-Wert-Paare, für die Sie das andere Sprite mit dem kleinen grünen Quadrat verwenden können.

# Anhang B

## Bei Verwendung von Scratch 1.4

# Die Einrichtung




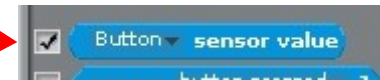
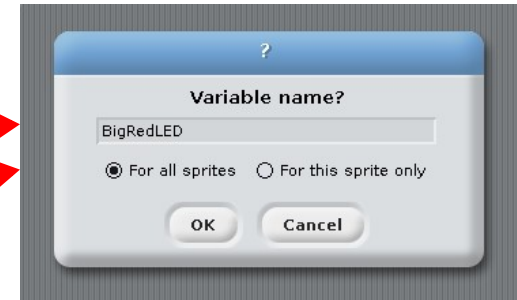
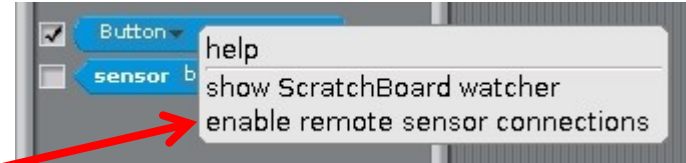


# Definieren der Konfigurationsdatei

- Nichts neues, Sie können die gleiche Konfigurationsdatei für Scratch 1.4 wie für Scratch 2 verwenden.

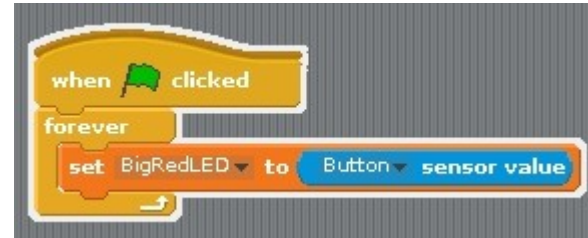
# Erstellen Sie das Scratch-Programm

- Start Scratch  → Entwicklung → Scratch
- Unterstützung für Netzwerksensoren einschalten (Rechtsklick auf Wert von Sensor)
- Erstellen Sie die Variable BigRedLED, die allen Sprites zur Verfügung steht
- Machen Sie die Variable sichtbar (kreuzen Sie das Kästchen an)
- Machen Sie den Button-Sensor sichtbar
- Speichern Sie die Datei auf dem Desktop.



# Programm in Scratch 1.4

- Stellen Sie dieses Programm auf "Scratch", wodurch die rote LED leuchtet, wenn die Taste losgelassen wird.
- Testen Sie, ob es funktioniert.



# Anhang B

## Grundlegende Elektronik

# Hinzugefügt werden

# Ende des Anfängerworkshops